Claims

1. A computer-implemented, incremental process for executing an application servo in a client device based on a specified set of matching criteria, the process comprising the steps of:

selecting a servo to provide services;

identifying a datasource associated with the selected servo;

initializing an execution context tree structure by creating a root node of the context tree associated with an initial instruction of the servo;

choosing a context of the context tree that satisfies the matching criteria;

executing an instruction of the servo associated with the chosen context;

responsive to said executing step, creating zero or more new child contexts in the context tree, each new child context including content defining a current internal evaluation state of the process; and

repeating said choosing, executing and creating steps over subsequent instructions of the servo until no context satisfies the matching criteria.

2. A process according to claim 1 wherein the content of the child context includes:

a pointer to an element within the selected servo; and

a pointer that identifies a current data context by pointing into a source tree.

3. A process according to claim 1 wherein the content of the child context includes:

a reference to a parent context;

an ordered, potentially sparse, list of pointers to zero or more child contexts; and definitions for any symbols introduced by the context.

4. A process according to claim 1 further including, responsive to said executing step, creating zero or more child spacers in the context tree representing unmaterialized child contexts; and wherein said choosing a context includes choosing either a context or a spacer.

5. A process according to claim 4 wherein the context tree is implemented using a relative b-tree structure, and each spacer is reflected in an interior node entry in the relative b-tree structure to facilitate searching unmaterialized contexts.

6. A process according to claim 1 wherein the b-tree node entry includes a field to track a linear value associated with a graphical display output object.

45

7. A process according to claim 1 wherein the process creates and maintains both the context tree and a geometry tree, the geometry tree representing the spatial structure of a predetermined graphical user interface.

8. A process according to claim 1 wherein the servo is defined using a servo definition language that references XML schema definitions as its core vocabulary.

9. A process according to claim 8 wherein the servo definition language comprises:

application data schema;

transformation rules; and

opportunity rules.

10. A servo definition language for defining a distributed application that supports disconnected operation, the language comprising the following types of rules:

application data schema;

transformation rules;

transaction handling rules; and

interface object specifications.

11. A servo definition language according to claim 10 further comprising access rules.

12. A servo definition language according to claim 10 further comprising opportunity rules to realize automatic extension or integration of servos through opportunity-based linking of an interface component representing an instance of a schema fragment to a template.

13. A servo definition language according to claim 12 wherein the template specifies at least one of a transformation rule, a transaction handling rule and an interface object specification.

14. A servo definition language according to claim 10 and further comprising an abstract interface object definition.

15. A servo definition language according to claim 10 wherein the application data schema comprises an XML-based schema.

16. A servo definition language according to claim 10 defined using XML schema definitions XSD as the core vocabulary.

17. A servo definition language according to claim 10 including a view element for selecting a group of the said transformation rules to define at least a part of an output interface.

18. A servo definition language according to claim 10 including a storage declaration element that enables an author to reserve and name persistent storage for use by the servo and any other servos authorized to access the corresponding data.

19. A servo definition language according to claim 18 wherein the storage declaration element includes a locally scoped name for a corresponding storage tree and identifies a schema to which the storage tree must conform.

20. A servo for realizing a predetermined function on a mobile client device having a user interface and utilizing at least intermittent communication with a server environment, the servo comprising:

a series of transformation elements expressed in a predetermined servo description language (SDL) consistent with a predetermined SDL schema;

the servo transformation elements including a data schema declaration for defining a data type on which the servo can be executed;

and the servo transformation elements further including at least two transformation rules, each rule for transforming selected elements of the data type defined by the data schema from an input document so as to provide output to the client device user interface; and further wherein

the servo transformation elements include at least one view declaration for selecting a group of the said transformation rules so as to define at least a part of an output interface realizable on the client device user interface.

21. A servo according to claim 20 wherein the servo is declared using a standard XSLT vocabulary extended so as to include the view declaration element.

22. A servo according to claim 20 wherein the servo transformation elements include at least one storage declaration element for declaration of a persistent storage tree.

23. A servo according to claim 22 wherein the storage declaration element includes a name attribute for defining a locally scoped name for the corresponding storage tree.

24. A servo according to claim 23 wherein the storage declaration element includes identification of a predetermined storage schema to which the corresponding storage tree conforms.

47

25. A servo according to claim 24 wherein the storage declaration element includes at least one attribute directed to specification of a lifetime, an update policy or a destruction policy associated with the corresponding storage tree.

26. A servo according to claim 24 wherein the predetermined storage schema is defined in an extensible stylesheet language.

27. A servo according to claim 24 wherein the predetermined storage schema is defined within the servo.

28. A servo according to claim 24 wherein the predetermined storage schema is defined by an XML schema definition.

29. A servo according to claim 24 including an opportunity element to realize automatic extension of the servo through opportunity-based linking of an interface component representing an instance of a schema fragment to a template wherein the template specifies at least one of a transformation rule, a transaction handling rule and an interface object specification.

30. A servo according to claim 24 wherein the servo is declared using a standard XSLT vocabulary extended so as to include the storage declaration element.

31. A computing platform for distributed mobile applications comprising a computer-implemented server environment for interaction with a mobile client machine over a potentially intermittent communication channel, the server environment including:

a messaging service for managing potentially intermittent communication between the client machine and the server environment;

a location broker component for locating a database associated with an identifier contained in a request communicated from the client machine;

a session manager service for managing an interactive session with the client machine;

an interpreter instance for processing a request assigned to it in a current session;

storage for at least one servo, data and schema; and

a registry for managing meta data associated with servos and schemata;

wherein both the servo and the data are expressed in a markup language.

32. A computing platform according to claim 31 wherein the stored servo is declared in a servo definition language (SDL) defined by a schema that references the XSL vocabulary; and the SDL includes a means for discovering a session environment; and

48

wherein the session environment includes indicia (attributes) of a physical location of the mobile client machine, identification of the mobile client hardware, and current status of the communication channel.

33. A computing platform according to claim 31 wherein the SDL defines a storage declaration instruction, and application data is available for reference in the SDL as a named input tree using a name defined in a storage declaration instruction in the servo.

34. A computing platform according to claim 33 wherein the SDL defines the following types of rules:

application data schema;

transformation rules;

transaction handling rules; and

interface object specifications.

35. A computing platform according to claim 34 wherein the SDL further defines opportunity rules to realize automatic extension a servo through opportunity-based linking of an interface component representing an instance of a schema fragment to a template.

36. A computing platform according to claim 34 wherein the SDL further defines opportunity rules to realize automatic integration of multiple servos through opportunity-based linking of an interface component representing an instance of a schema fragment to a template.

37. A computing platform according to claim 35 wherein the SDL specifies at least one of a transformation rule, a transaction handling rule and an interface object specification.

38. A computing platform according to claim 35 wherein the SDL further includes a view element for selecting a group of transformation rules to define at least a part of an output interface.

39. A computing platform according to claim 38 wherein the server environment further includes a schema translator for translating data from one schema to another.

40. A computing platform according to claim 38 wherein the registry includes at least one of the following kinds of metadata:

a schema public identifier with the corresponding schema representation;

a schema translator with a corresponding directed pair of schema public identifiers including flags for whether the translation loses information; and

49

a datasource public identifier with corresponding datasource declaration.